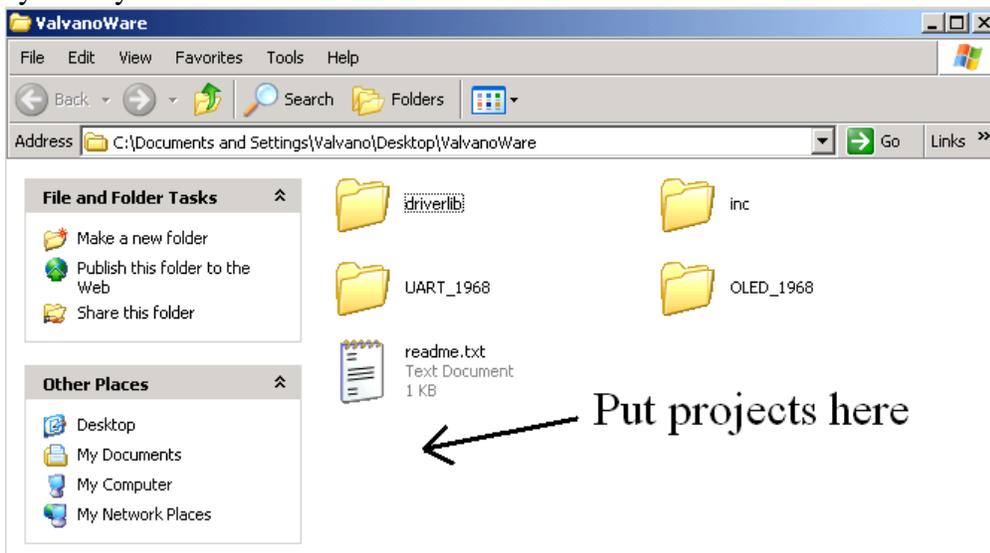


Creating a Project:

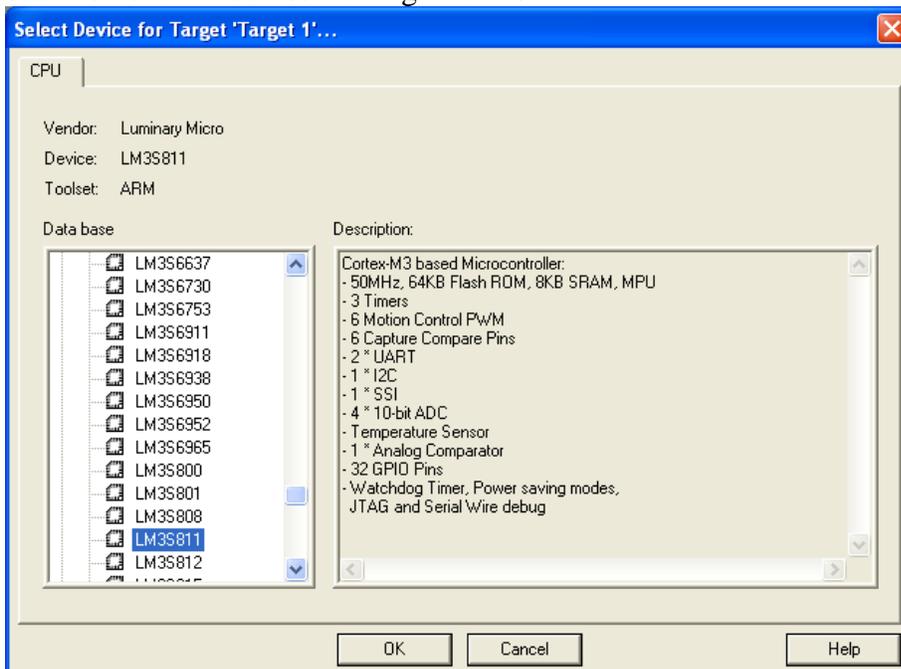
0) Start up Keil uVision4

1) Under the “Project” menu, select “New uVision Project...”

2) Navigate to a folder where the project will be created, and give it a name. Click “OK” to continue. If you have downloaded ValvanoWare, you can put the new project in there. For most systems you will need access to the **driverlib** and the **inc** folders.



3) Click the [+] next to the manufacturer of your device and find your device from the resulting list. For example, click “[+] Texas Instruments” and then click on “LM3S1968”. For older versions of the compiler, Stellaris microcontrollers can be found under “Luminary Micro”. Your selection should look something like this:



Click “OK” to continue.

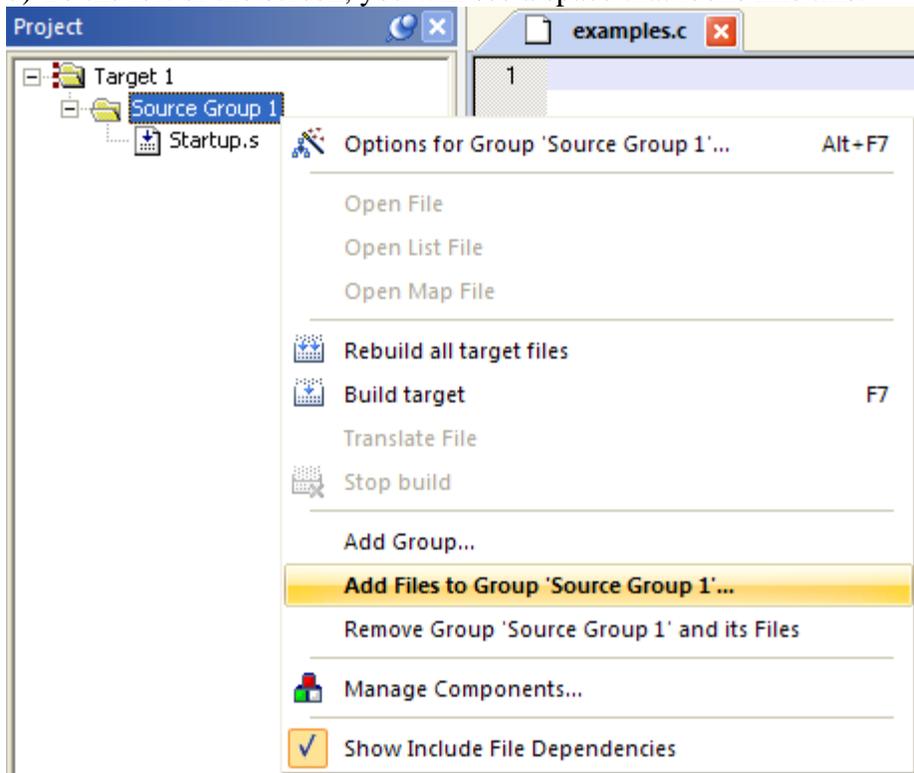
4) Click “Yes” when asked to copy Luminary Startup Code to Project Folder and Add File to Project.



5) Under the “File” menu, select “New...” to create a blank text space.

6) Under the “File” menu, select “Save As...” to save your new text space. Give it the extension “.c” to begin programming or “.txt” to provide documentation.

7) To the left of the screen, you will see a space that looks like this:



Right-click on “Source Group 1” and select “Add Files to Group ‘Source group 1’...”.

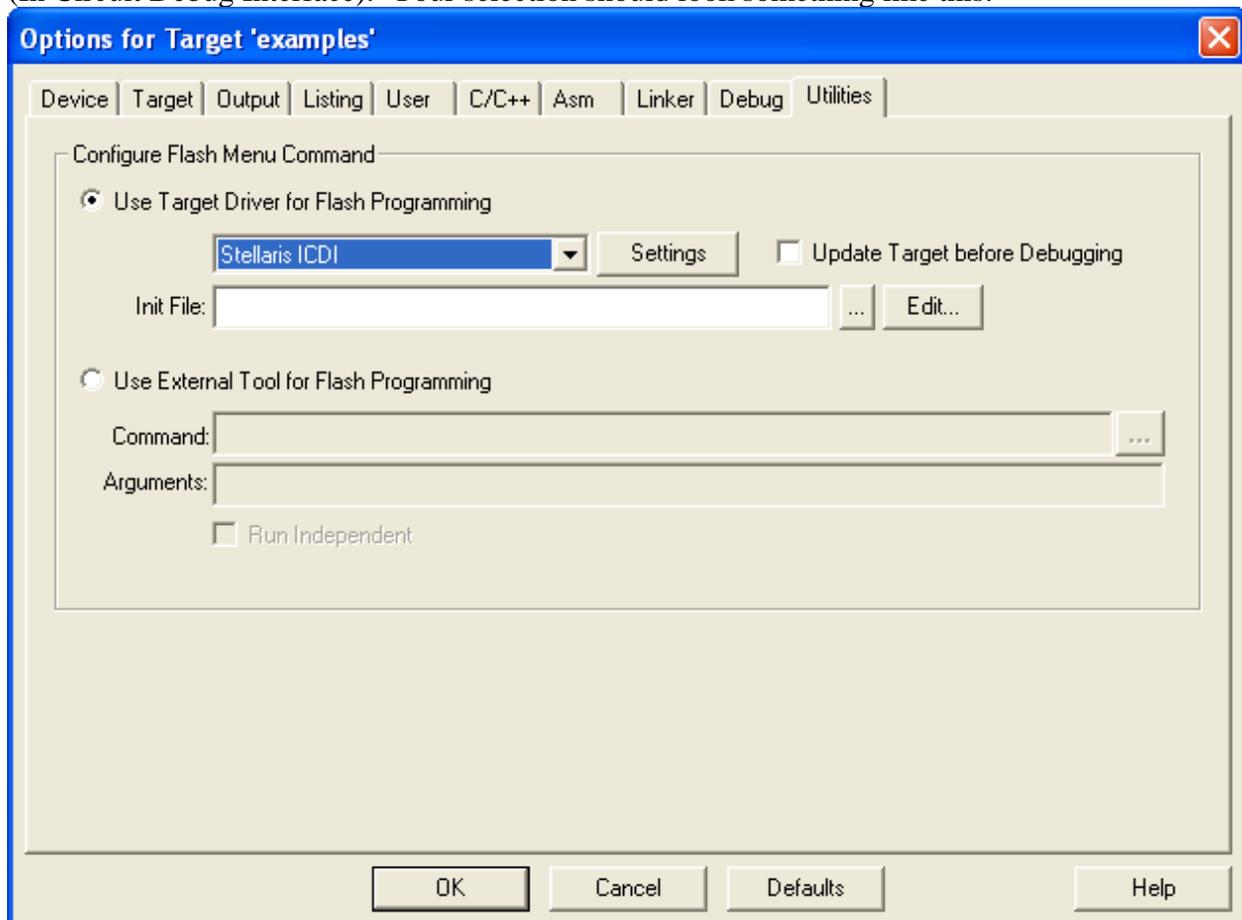
8) Select the file(s) that you saved in Step 5 and click “Add” for each one. Click “Close” when finished. If you want, you can re-name “Target 1” and “Source Group 1” by selecting them and pressing F2. You can also create additional sub-groups by right-clicking on “Target 1” and selecting “Add Group...”. The new group can be populated with files and re-named in the same way as “Source Group 1”. For example, you may want to add library files to your Project. Libraries give you easy access to common, useful functions without needing to write them yourself. Right-click on “Target 1”, select “Add Group...”, and give the new group a descriptive

name such as “Library”. Right-click on the “Library” group and select “Add Files to Group ‘Library’...”. Navigate to the folder that contains library files, such as “...driverlib\rvmdk\driverlib.lib” and add them to this group.

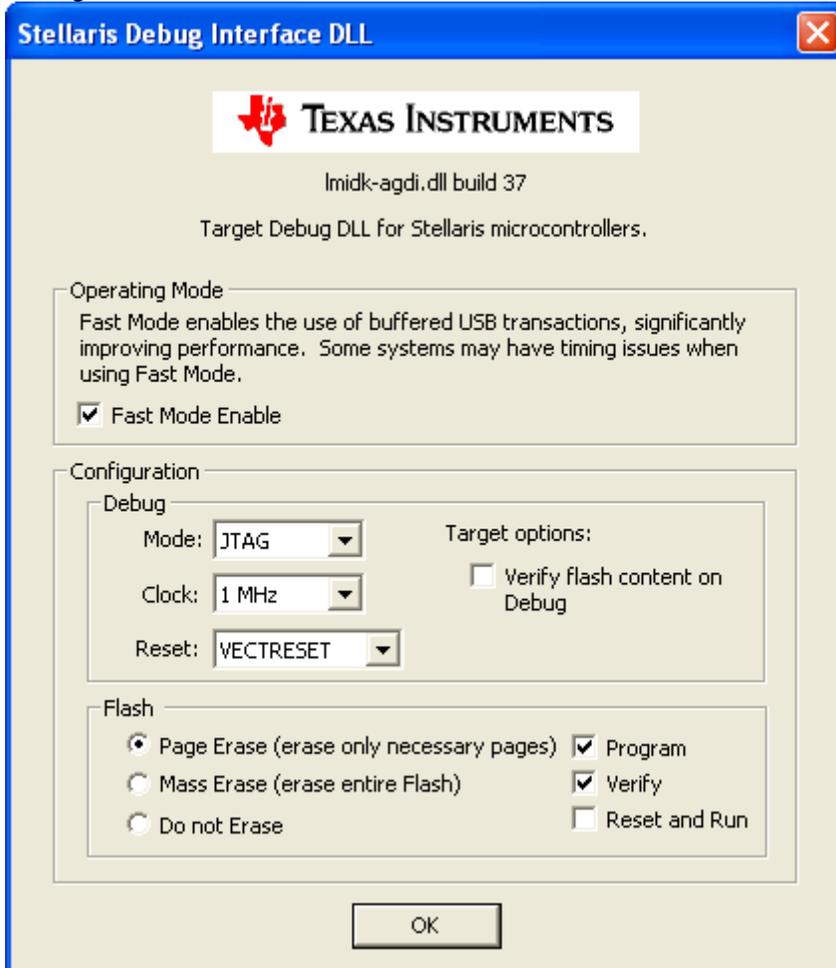
In ValvanoWare, the **driverlib.lib** is in the **driverlib\rvmdk** directory

9) Under the “Flash” menu, select “Configure Flash Tools...”.

10) In the “Options” window that pops up, go to the “Utilities” sub-tab. Select the hardware that will be used to program the flash from the pull-down menu. For example, select “Stellaris ICDI” (In Circuit Debug Interface). Your selection should look something like this:

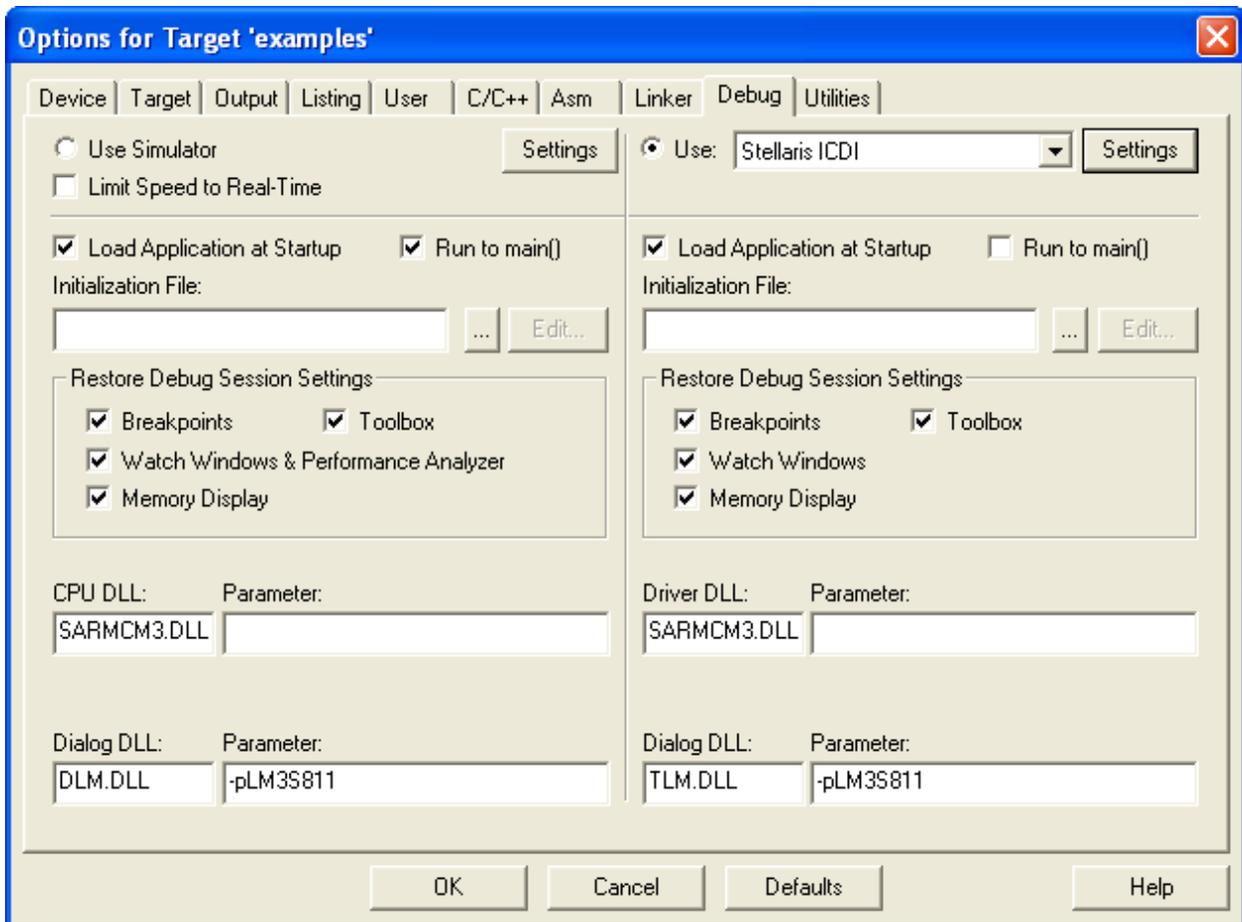


11) Click the button that says “Settings” next to the pull-down menu. Change the settings to this configuration:



Click “OK” to continue.

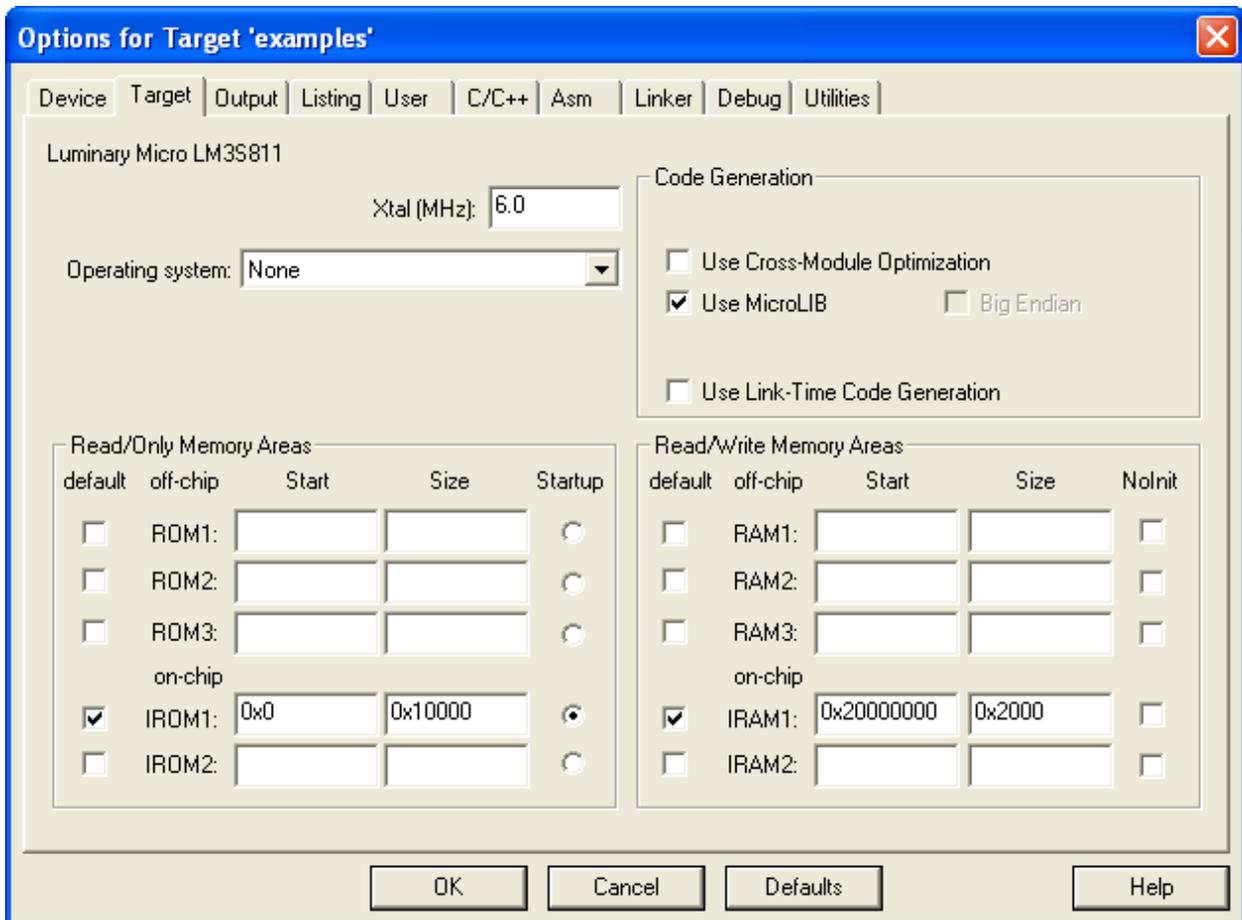
12) While still in the “Options” window, go to the “Debug” sub-tab. We will be programming actual hardware and not just running the simulator. Therefore, select the button that says “Use: _____”, which deactivates the button that says “Use Simulator”. Fill in the blank by selecting the hardware that will be used to program the flash from the pull-down menu just like you did in Step 10. Continuing the example: select “Stellaris ICD1”. Your selection should look something like this:



13) While still in the “Options” window, go to the “C/C++” sub-tab. If you want to use external files, the compiler must be able to find them. You can put the entire path in the “#include” statements, but your program will only work with one very specific directory structure. A better solution is to give the compiler a general idea of where to look for files using the “Include Paths”, and organize your files in a consistent and meaningful way. To begin looking for include files one and three directories up from the directory that contains “__uvproj”, change the settings to this configuration:

Include paths should have at a minimum:

```
..;..\..\.;..\driverlib;..\inc;
```

Click “OK” to continue. Setup should be complete, so you are now ready to program.

Problems I’ve encountered and how to fix them:

“Contents mismatch[*sic*] at: <memory location> !
<repeated many times>

Too many errors to display !”

Memory may not be correctly over-written. See Steps 10 and 11.

“Error: Flash Download failed - "Cortex-M3””

The flash downloader is programmed with incorrect attributes for your target device. See step 14.

Either nothing happens or a fault occurs after activating a port.

Enabling a peripheral requires a few additional cycles, so other instructions should come between the instruction that activates the port and the instructions that initialize it.